K 7

```
RRRRRRRRRRRR     UUU         UUU  NNN         NNN     OOOOOOOOO     FFFFFFFFFFFFFFF  FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN         NNN     OOOOOOOOO     FFFFFFFFFFFFFFF  FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN         NNN     OOOOOOOOO     FFFFFFFFFFFFFFF  FFFFFFFFFFFFFFF
RRR         RRR  UUU         UUU  NNN         NNN  OOO         OOO  FFF              FFF
RRR         RRR  UUU         UUU  NNN         NNN  OOO         OOO  FFF              FFF
RRR         RRR  UUU         UUU  NNN         NNN  OOO         OOO  FFF              FFF
RRR         RRR  UUU         UUU  NNNNNN      NNN  OOO         OOO  FFF              FFF
RRR         RRR  UUU         UUU  NNNNNN      NNN  OOO         OOO  FFF              FFF
RRR         RRR  UUU         UUU  NNNNNN      NNN  OOO         OOO  FFF              FFF
RRRRRRRRRRRR     UUU         UUU  NNN    NNN  NNN  OOO         OOO  FFFFFFFFFFFF     FFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN    NNN  NNN  OOO         OOO  FFFFFFFFFFFF     FFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN    NNN  NNN  OOO         OOO  FFFFFFFFFFFF     FFFFFFFFFFFF
RRR   RRR        UUU         UUU  NNN      NNNNNN  OOO         OOO  FFF              FFF
RRR   RRR        UUU         UUU  NNN      NNNNNN  OOO         OOO  FFF              FFF
RRR   RRR        UUU         UUU  NNN      NNNNNN  OOO         OOO  FFF              FFF
RRR      RRR     UUU         UUU  NNN         NNN  OOO         OOO  FFF              FFF
RRR      RRR     UUU         UUU  NNN         NNN  OOO         OOO  FFF              FFF
RRR      RRR     UUU         UUU  NNN         NNN  OOO         OOO  FFF              FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN         NNN     OOOOOOOOO     FFF              FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN         NNN     OOOOOOOOO     FFF              FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN         NNN     OOOOOOOOO     FFF              FFF
```

```
_$2

Sym
---
NDX
NDX
NUM
NUM
OPE
OUT
PAC
PAC
PAC
PAC
PAC
PAC
PAC
PAD
PAG
PAG
PAG
PAG
PAG
PAG
PAG
PER
PUT
RCO
RIN

RLI
RNO
RNO
RTY
SAV
STR
STR
STR
STR
STR
STR

STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
```

```
**FILE**ID**FOOOUT                                    N 11








FFFFFFFFFF    OOOOOO      OOOOOO      OOOOOO    UU      UU  TTTTTTTTTT
FFFFFFFFFF    OOOOOO      OOOOOO      OOOOOO    UU      UU  TTTTTTTTTT
FF          OO      OO  OO      OO  OO      OO    UU      UU      TT
FF          OO      OO  OO      OO  OO      OO    UU      UU      TT
FF          OO      OO  OO      OO  OO      OO    UU      UU      TT
FF          OO      OO  OO      OO  OO      OO    UU      UU      TT
FFFFFFFF    OO      OO  OO      OO  OO      OO    UU      UU      TT
FFFFFFFF    OO      OO  OO      OO  OO      OO    UU      UU      TT
FF          OO      OO  OO      OO  OO      OO    UU      UU      TT
FF          OO      OO  OO      OO  OO      OO    UU      UU      TT
FF          OO      OO  OO      OO  OO      OO    UU      UU      TT          ....
FF          OO      OO  OO      OO  OO      OO    UU      UU      TT          ....
FF            OOOOOO      OOOOOO      OOOOOO      UUUUUUUUUU      TT          ....
FF            OOOOOO      OOOOOO      OOOOOO      UUUUUUUUUU      TT          ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
      0001  0 %TITLE 'performs outputting of footnotes'
      0002  0 MODULE fooout ( IDENT = 'V04-000'
    P 0003  0               %BLISS32[, ADDRESSING_MODE (EXTERNAL    = LONG_RELATIVE,
      0004  0                                           NONEXTERNAL = LONG_RELATIVE)]
      0005  0               ) =
      0006  1 BEGIN
      0007  1 !
      0008  1 !****************************************************************************
      0009  1 !*                                                                          *
      0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
      0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
      0012  1 !*  ALL RIGHTS RESERVED.                                                    *
      0013  1 !*                                                                          *
      0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
      0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
      0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
      0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
      0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
      0019  1 !*  TRANSFERRED.                                                            *
      0020  1 !*                                                                          *
      0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
      0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
      0023  1 !*  CORPORATION.                                                            *
      0024  1 !*                                                                          *
      0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
      0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
      0027  1 !*                                                                          *
      0028  1 !*                                                                          *
      0029  1 !****************************************************************************
      0030  1 !
      0031  1 !++
      0032  1 ! FACILITY:      DSR (Digital Standard RUNOFF) / DSRPLUS
      0033  1 !
      0034  1 ! ABSTRACT:      Puts out footnotes.
      0035  1 !
      0036  1 ! ENVIRONMENT:   Transportable
      0037  1 !
      0038  1 ! AUTHOR:        R.W.Friday       CREATION DATE: September, 1978
      0039  1
```

```
;   41          0040  1 %SBTTL 'Revision History'
;   42          0041  1 ! MODIFIED BY:
;   43          0042  1 !
;   44          0043  1 !       012     RER00012        Ron Randall     13-Apr-1983
;   45          0044  1 !               Fixed obscure bug with numbered footnotes.
;   46          0045  1 !
;   47          0046  1 !       011     RER00011        Ron Randall     06-Apr-1983
;   48          0047  1 !               For DSRPLUS:
;   49          0048  1 !               For numbered footnotes, a line of dashes is output first.
;   50          0049  1 !
;   51          0050  1 !       010     RER00010        Ron Randall     07-Mar-1983
;   52          0051  1 !               Global edit of all modules. Updated module names, idents,
;   53          0052  1 !               copyright dates. Changed require files to BLISS library.
;   54          0053  1 !--
;   55          0054  1
;    57        0055  1 %SBTTL 'Module Level Declarations'
;    58        0056  1 !
;    59        0057  1 ! INCLUDE FILES:
;    60        0058  1 !
;    61        0059  1 LIBRARY 'NXPORT:XPORT';                         ! XPORT Library
;    62        0060  1 REQUIRE 'REQ:RNODEF';                           ! RUNOFF variant definitions
;    63        0191  1
;    64      U 0192  1 %IF DSRPLUS %THEN
;    65      U 0193  1 LIBRARY 'REQ:DPLLIB';                           ! DSRPLUS BLISS Library
;    66        0194  1 %ELSE
;    67        0195  1 LIBRARY 'REQ:DSRLIB';                           ! DSR BLISS Library
;    68        0196  1 %FI
;    69        0197  1
;    70        0198  1 !
;    71        0199  1 ! EXTERNAL REFERENCES:
;    72        0200  1 !
;    73        0201  1 EXTERNAL LITERAL
;    74        0202  1     RINTES : UNSIGNED (8);
;    75        0203  1
;    76        0204  1 EXTERNAL LITERAL
;    77        0205  1     RNFILE;                                     !Error messages
;    78        0206  1
;    79        0207  1 EXTERNAL
;    80        0208  1     FNCT : FNCT_DEFINITION,
;    81        0209  1     FNISIZ : FN_INT_SIZE_DEFINITION,
;    82        0210  1     FOOREC : FOOREC_DEFINITION,
;    83        0211  1     GCA : GCA_DEFINITION,
;    84        0212  1     MRA : REF FIXED_STRING,
;    85        0213  1     PHAN : PHAN_DEFINITION,
;    86        0214  1     TSF : TSF_DEFINITION;
;    87        0215  1
;    88        0216  1 EXTERNAL
;    89        0217  1     FOOMRA : FIXED_STRING,                      !Substitute MRA.
;    90        0218  1     FOOTSF : VECTOR [TSF_SIZE];                 !Substitute TSF area
;    91        0219  1
;    92        0220  1 EXTERNAL
;    93        0221  1     FOHMRA,
;    94        0222  1     FOHTSF;
;    95        0223  1
;    96      U 0224  1 %IF DSRPLUS %THEN
;    97      U 0225  1 EXTERNAL
;    98      U 0226  1     FRA : FIXED_STRING;
;    99      U 0227  1
;   100      U 0228  1 EXTERNAL ROUTINE
;   101      U 0229  1     LOUT;
;   102        0230  1 %FI
;   103        0231  1
;   104        0232  1 EXTERNAL ROUTINE
;   105        0233  1     ERMS,
;   106        0234  1     FOOFIL,
;   107        0235  1     OUTLIN,
;   108        0236  1     OUTPAS;
;   109        0237  1
```

```
1 %SBTTL 'body of routine'
1 GLOBAL ROUTINE fooout (OUTPUT_COUNT) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       FOOOUT outputs as many footnotes as it is requested to do.  For each
1 !       footnote, all it does is read the appropriate number of TSF/MRA pairs
1 !       from the footnote file, and calls OUTLIN to process the information.
1 !       The number of TSF/MRA sets for any particular footnote is stored at
1 !       FNISIZ [n].
1 !
1 !       FOOOUT also takes care of index entries that it finds.  The transaction
1 !       numbers are remapped into a different set, so that they get merged with
1 !       entries from the main body of the document.
1 !
1 !       After FOOOUT has output the footnotes it's supposed to output, it
1 !       leaves the file open.  It does not close the file, or attempt to clean
1 !       it up in any way.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       OUTPUT_COUNT - Number of footnotes to be output.
1 !
1 ! IMPLICIT INPUTS:        None
1 !
1 ! IMPLICIT OUTPUTS:       None
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:       None
1 !
1 ! SIDE EFFECTS:           None
1 !--
1
2     BEGIN
2     LOCAL
2         COUNT,
2         FOO_FIRST_XTN,
2         HOLD_PAGING,
2         MAP_FIRST_XTN,
2         STATUS;
2
2     IF  .OUTPUT_COUNT LEQ 0       ! If this is a meaningless call,
2     THEN
2         RETURN                    !   ignore it.
2     ELSE                          ! Otherwise, check the validity of the request
3         BEGIN
3
3         IF  .OUTPUT_COUNT GTR .FNCT_READY !  If the user asked for more
3         THEN                              !  footnotes to be output than are
4             BEGIN                         !  ready to be output,
4             ERMS (RNFILE, CH$PTR (UPLIT ('FOOOUT')), 6); ! report the error
4             COUNT = .FNCT_READY;          !      and use the real count instead.
4             END
3         ELSE                              !  Otherwise, setup to output the #
3             COUNT = .OUTPUT_COUNT;        !  the user requested.
3
```

```
;   168          0295  2          END;
;   169          0296  2
;   170          0297  2      ! NOTE: The change to not using pool for footnotes necessitated changing a
;   171          0298  2      ! test for .FNPOOL. This change MAY have reintroduced an old bug from
;   172          0299  2      ! either # 020 or # 025.  Check for it and beware.
;   173          0300  2      !  RUNOFF V2(1), FN IDENT 008;  9-2-1980. kad.
;   174          0301  2
;   175          0302  2      IF  FOOFIL (FOO_OPIN) NEQ FOO_NORMAL
;   176          0303  2      THEN
;   177          0304  3          BEGIN   ! Can't read back footnotes: give up and forget them.
;   178          0305  3          FOOFIL (FOO_CLOS);
;   179          0306  3          FOOFIL (FOO_DELE);
;   180          0307  3          FNCT_COUNT   = 0;
;   181          0308  3          FNCT_READY   = 0;
;   182          0309  3          FNCT_WAITING = 0;
;   183          0310  3          FNCT_OLD     = 0;
;   184          0311  3          RETURN
;   185          0312  2          END;
;   186          0313  2
;   187          0314  2      !The footnote file was opened successfully.
;   188          0315  2      FNCT_EXPANDING = TRUE;                          !Signal the fact that footnotes are being expanded.
;   189          0316  2      HOLD_PAGING    = .PHAN_PAGING;                  !Go into non-paging mode to
;   190          0317  2      PHAN_PAGING    = FALSE;                         !avoid spurious page breaks.
;   191          0318  2      !Use the footnote working areas to save results in.
;   192          0319  2      FOHTSF = .TSF;                                  !Save current status.
;   193          0320  2      FOHMRA = .MRA;
;   194          0321  2      TSF = FOOTSF;                                   !Switch to other working area.
;   195          0322  2      MRA = FOOMRA;
;   196          0323  2      FOO_FIRST_XTN = 0;                              !Initialize transaction number
;   197          0324  2      MAP_FIRST_XTN = 0;
;   198          0325  2
;   199        U 0326  2 %IF DSRPLUS %THEN
;   200        U 0327  2      !
;   201        U 0328  2      ! If outputting numbered footnotes, first output a line of dashes.
;   202        U 0329  2      !
;   203        U 0330  2      IF  .FNCT_NUMBERING
;   204        U 0331  2      THEN
;   205        U 0332  2          BEGIN
;   206        U 0333  2          !
;   207        U 0334  2          ! Initialize fixed string.
;   208        U 0335  2          !
;   209        U 0336  2          FS_INIT (MRA);
;   210        U 0337  2          !
;   211        U 0338  2          ! Write dashes into the fixed string.
;   212        U 0339  2          !
;   213        U 0340  2          INCR I FROM 1 TO 15 DO
;   214        U 0341  2              FS_WCHAR (MRA, %C'-');
;   215        U 0342  2
;   216        U 0343  2          !
;   217        U 0344  2          ! Update the appropriate counts.
;   218        U 0345  2          !
;   219        U 0346  2          TSF_INT_HL = 15;
;   220        U 0347  2          TSF_EXT_HL = 15;
;   221        U 0348  2          TSF_INT_VL = 0;
;   222        U 0349  2          !
;   223        U 0350  2          ! Put out the line as OUTLIN does.
;   224        U 0351  2          !
;   225       U 0352  2               FS_INIT (FRA);
;   226       U 0353  2               GCA_LINE_PEND = 0;
;   227       U 0354  2               LOUT ();
;   228       U 0355  2               END;
;   229         0356  2   %FI
;   230         0357  2
;   231         0358  2           !Now actually read back the records.
;   232         0359  2           INCR I FROM 0 TO (.COUNT - 1) DO
;   233         0360  2           INCR J FROM 1 TO .FNISIZ [.FNCT_OLD + .I] DO                  !(J is a dummy counter, not really used anyw
;   234         0361  3               BEGIN
;   235         0362  3               !Determine which kind of record is coming: a TSF/MRA pair, or a "pass through"
;   236         0363  3               !record.
;   237         0364  3               STATUS = FOOFIL (FOO_RHDR);
;   238         0365  3
;   239         0366  3               IF  .STATUS NEQ FOO_NORMAL
;   240         0367  3               THEN
;   241         0368  3                   !Exit loop if record can't be read.
;   242         0369  3                   EXITLOOP;
;   243         0370  3
;   244         0371  3               !Is what's coming a TSF/MRA pair, or is it "pass through" information?
;   245         0372  3               IF  .FOOREC_MAJOR_TYPE EQL FOOREC_MAJ_PASS
;   246         0373  3               THEN
;   247         0374  3                   !Direct "pass through" information to the appropriate destination
;   248         0375  4                   BEGIN
;   249         0376  4                   !First read the "pass through" information
;   250         0377  4                   STATUS = FOOFIL (FOO_PREAD);
;   251         0378  4
;   252         0379  4                   !Make sure the read worked.
;   253         0380  4                   IF  .STATUS NEQ FOO_NORMAL
;   254         0381  4                   THEN
;   255         0382  4                       !Information couldn't be read.  !Quit processing.
;   256         0383  4                       EXITLOOP;
;   257         0384  4
;   258         0385  4                   !Map transaction number, if any.
;   259         0386  4                   IF  .FOOREC_XTN_PTR GTR 0
;   260         0387  4                   THEN
;   261         0388  4                       !This pass thru record contains a transaction number.  Remap it into something else.
;   262         0389  5                       BEGIN
;   263         0390  5                       BIND
;   264         0391  5                           PASS_THRU_RECORD = .FOOREC_ADDRESS : VECTOR;
;   265         0392  5                           !Note:  Never change transaction numbers that are zero.
;   266         0393  5
;   267         0394  5                       IF  .PASS_THRU_RECORD [.FOOREC_XTN_PTR - 1] NEQ 0
;   268         0395  5                       THEN
;   269         0396  6                           BEGIN
;   270         0397  6
;   271         0398  6                           IF  .FOO_FIRST_XTN EQL 0
;   272         0399  6                           THEN
;   273         0400  6                               !Establish start of mapping from temporarily assigned
;   274         0401  6                               !transaction numbers to the normal set.
;   275         0402  7                               BEGIN
;   276         0403  7                               MAP_FIRST_XTN = .GCA_NORMAL_XTN;          !Establish Y(1).
;   277         0404  7                               FOO_FIRST_XTN = .PASS_THRU_RECORD [.FOOREC_XTN_PTR - 1];         !Establish X(1).
;   278         0405  6                               END;
;   279         0406  6
;   280         0407  6                           !Translate the temporary transaction number assigned
;   281         0408  6                           !to this pass thru record.
```

```
            !This is done by simply substituting the current Y(1)
            !(which is GCA_NORMAL_XTN) for the given X(1).
            !Then, GCA_NORMAL_XTN is bumped, thereby giving a
            !new Y(1) for the next such record, if any.
            PASS_THRU_RECORD [.FOOREC_XTN_PTR - 1] = .GCA_NORMAL_XTN;
            GCA_NORMAL_XTN = .GCA_NORMAL_XTN + 1;
            END

        END;

    OUTPAS (.FOOREC_RECORD_SIZE, .FOOREC_ADDRESS, .FOOREC_XTN_PTR,
                .FOOREC_MINOR_TYPE)
    END
ELSE
    !It's TSF/MRA information to be processed, so get it and process it.
    BEGIN
    FOOFIL (FOO_READ);

    !Make sure the read worked.
    IF  .STATUS NEQ FOO_NORMAL
    THEN
        !Information couldn't be read.  !Quit processing.
        EXITLOOP;

    !See if this record has anything to do with indexing.
    !If so, the record either contains an entry for the index,
    !or else it contains text that refers to something in the
    !index (or the index refers to it, depending on your point of
    !view.
    !
    !Records containing index entries always occur before the
    !record to which they apply; each record describes one
    !entry.  Further, each index entry has associated with it
    !a unique number, TSF_FIRST_XTN.  These transaction numbers
    !are strictly increasing from record to record.  Assume that
    !there are n such records, and let the transaction numbers
    !be designated by X(1), X(2), .... X(n).
    !
    !Records which refer to index entries come after the index
    !entry records; they are chained to the index entries by
    !having the transaction numbers recorded in the corresponding
    !TSF.  TSF_FIRST_XTN will have the value X(1), and TSF_LAST_XTN
    !will have the value X(n).
    !
    !The problem is that X(1).....X(n) cannot be directly used
    !by the routine ASGXTN (which is in the module XTN).  That
    !is because the association of transaction numbers to pages
    !requires that the transaction numbers passing through ASGXTN
    !be steadily increasing, with no gaps, FOR THE ENTIRE FILE.
    !Now, X(1)......X(n) are steadily increasing, but only
    !for the footnote.  However, they can be mapped into the
    !"correct" set, which is accomplished here.
    !Note that the new set will be designated by Y(1)......Y(n).
    !Further, Y(1) is guaranteed to be GCA_NORMAL_XTN.
    IF  .TSF_FIRST_XTN NEQ 0
    THEN

```

```
;  339      0466  4          !This record has something to do with indexing.
;  340      0467  4          IF   .TSF_INDEX
;  341      0468  4          THEN
;  342      0469  4              !Process index entry.
;  343      0470  5              BEGIN
;  344      0471  5
;  345      0472  5              IF   .FOO_FIRST_XTN EQL 0
;  346      0473  5              THEN
;  347      0474  5                  !Establish start of mapping from temporarily assigned
;  348      0475  5                  !transaction numbers to the normal set.
;  349      0476  6                  BEGIN
;  350      0477  6                  MAP_FIRST_XTN = .GCA_NORMAL_XTN;       !Establish Y(1).
;  351      0478  6                  FOO_FIRST_XTN = .TSF_FIRST_XTN; !Establish X(1).
;  352      0479  5                  END;
;  353      0480  5
;  354      0481  5              !Translate the temporary transaction number assigned
;  355      0482  5              !to this index entry into a normal one.
;  356      0483  5              !This is done by simply substituting the current Y(1)
;  357      0484  5              !(which is GCA_NORMAL_XTN) for the given X(1).
;  358      0485  5              !Then, GCA_NORMAL_XTN is bumped, thereby giving a
;  359      0486  5              !new Y(1) for the next index entry, if any.
;  360      0487  5              TSF_FIRST_XTN  = .GCA_NORMAL_XTN;
;  361      0488  5              TSF_LAST_XTN   = .GCA_NORMAL_XTN;
;  362      0489  5              GCA_NORMAL_XTN = .GCA_NORMAL_XTN + 1;
;  363      0490  5              END
;  364      0491  4          ELSE
;  365      0492  4              !Process reference to index entry.
;  366      0493  4              !Notice that earlier, the starting X(1) and Y(1) were
;  367      0494  4              !established.  But that was for the start of the
;  368      0495  4              !footnote, not for the current record.  In the meantime,
;  369      0496  4              !several sets of index entries and references may have
;  370      0497  4              !gone through here.  And there is no way to separate them.
;  371      0498  4              !That means you can't simply use MAP_FIRST_XTN as the
;  372      0499  4              !new Y(1).
;  373      0500  5              BEGIN
;  374      0501  5              LOCAL
;  375      0502  5                  XTN_OFFSET;
;  376      0503  5
;  377      0504  5              !Compute the gap between the X(1) on this record
;  378      0505  5              !and that at the start of the footnote processing.
;  379      0506  5              !XTN_OFFSET, the result, will be zero for the very first
;  380      0507  5              !record that comes through.
;  381      0508  5              XTN_OFFSET = .TSF_FIRST_XTN - .FOO_FIRST_XTN;
;  382      0509  5
;  383      0510  5              !Now, compute the desired Y(n).
;  384      0511  5              !The value MAP_FIRST_XTN+XTN_OFFSET gets you the Y(1)
;  385      0512  5              !you need.  The value TSF_LAST_XTN - TSF_FIRST_XTN gets
;  386      0513  5              !you "n".
;  387      0514  5              TSF_LAST_XTN = .MAP_FIRST_XTN + .XTN_OFFSET + (.TSF_LAST_XTN - .TSF_FIRST_XTN);
;  388      0515  5
;  389      0516  5              !This is left as an exercise for the reader.
;  390      0517  5              TSF_FIRST_XTN = .MAP_FIRST_XTN + .XTN_OFFSET;
;  391      0518  5
;  392      0519  5              !As an additional exercise, prove that it's
;  393      0520  5              !not necessary to clear FOO_FIRST_XTN now, or later;
;  394      0521  5              !Further, show that it doesn't matter whether or
;  395      0522  5              !not you do clear it.  Using the results of these
;  396          0523  5                     !exercises simplify this block of code.
;  397          0524  4                     END;
;  398          0525  4
;  399          0526  4             !Write out the record.
;  400          0527  4             OUTLIN (0);                             !(Parameter is meaningless in this case).
;  401          0528  3             END;
;  402          0529  3
;  403          0530  2         END;
;  404          0531  2
;  405          0532  2     TSF = .FOHTSF;                                  !Restore interrupted status.
;  406          0533  2     MRA = .FOHMRA;
;  407          0534  2     PHAN_PAGING = .HOLD_PAGING;
;  408          0535  2     FNCT_READY  = .FNCT_READY - .COUNT;             !Count off the footnotes that have been output.
;  409          0536  2     FNCT_COUNT  = .FNCT_COUNT - .COUNT;             !Count off the footnotes that have been output.
;  410          0537  2     FNCT_OLD    = .FNCT_OLD + .COUNT;               !Update count of obsolete footnotes.
;  411          0538  2     FNCT_EXPANDING = FALSE;                         !No longer expanding footnotes
;  412          0539  2
;  413          0540  2     !Get rid of the work files if there are no more footnotes to be output.
;  414          0541  2     IF  .FNCT_COUNT LEQ 0
;  415          0542  2     THEN
;  416          0543  2         !All footnotes have been output.
;  417          0544  3         BEGIN
;  418          0545  3         FOOFIL (FOO_CLOS);                          !Close the file
;  419          0546  3         FOOFIL (FOO_DELE);                          !and delete it.
;  420          0547  3         FNCT_OLD     = 0;                           !And clean up the tables a bit.
;  421          0548  3         FNCT_READY   = 0;
;  422          0549  3         FNCT_WAITING = 0;
;  423          0550  2         END;
;  424          0551  2
;  425          0552  1     END;                                            !End of FOOOUT


                                                            .TITLE  FOOOUT performs outputting of footnotes
                                                            .IDENT  \V04-000\

                                                            .PSECT  $PLIT$,NOWRT,NOEXE,2

                    00  00  54  55  4F  4F  4F  46  00000 P.AAA:  .ASCII  \FOOOUT\<0><0>                              ;

                                                            .EXTRN  RINTES, RNFILE, FNCT
                                                            .EXTRN  FNISIZ, FOOREC, GCA
                                                            .EXTRN  MRA, PHAN, TSF, FOOMRA
                                                            .EXTRN  FOOTSF, FOHMRA, FOHTSF
                                                            .EXTRN  ERMS, FOOFIL, OUTLIN
                                                            .EXTRN  OUTPAS

                                                            .PSECT  $CODE$,NOWRT,2

                                      0FFC 00000            .ENTRY  FOOOUT, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-    ; 0239
                                                                    R11                                           ;
                    5B 00000000G   EF   9E 00002            MOVAB   FNCT+4, R11                                   ;
                    50       04    AC   D0 00009            MOVL    OUTPUT_COUNT, R0                              ; 0280
                                   01   14 0000D            BGTR    1$                                            ;
                                        04 0000F            RET                                                   ;
                    6B             50   D1 00010 1$:        CMPL    R0, FNCT+4                                    ; 0286
                                   1A   15 00013            BLEQ    2$                                            ;
                                   06   DD 00015            PUSHL   #6                                            ; 0289
```

```
                00000000'  EF  9F 00017         PUSHAB  P.AAA                     ;
                00000000G  8F  DD 0001D         PUSHL   #RNFILE                   ;
   00000000G EF            03  FB 00023         CALLS   #3, ERMS                  ;
             57            6B  D0 0002A         MOVL    FNCT+4, COUNT             ; 0290
                           03  11 0002D         BRB     3$                        ; 0286
             57            50  D0 0002F 2$:     MOVL    R0, COUNT                 ; 0293
                           01  DD 00032 3$:     PUSHL   #1                        ; 0302
   00000000G EF            01  FB 00034         CALLS   #1, FOOFIL                ;
             01            50  D1 0003B         CMPL    R0, #1                    ;
                           19  13 0003E         BEQL    4$                        ;
                           05  DD 00040         PUSHL   #5                        ; 0305
   00000000G EF            01  FB 00042         CALLS   #1, FOOFIL                ;
                           06  DD 00049         PUSHL   #6                        ; 0306
   00000000G EF            01  FB 0004B         CALLS   #1, FOOFIL                ;
                       FC  AB  7C 00052         CLRQ    FNCT                      ; 0307
                       08  AB  7C 00055         CLRQ    FNCT+12                   ; 0310
                               04 00058         RET                               ; 0304
          14 AB            01  D0 00059 4$:     MOVL    #1, FNCT+24               ; 0315
             5A 00000000G  FF  D0 0005D         MOVL    @PHAN+40, HOLD_PAGING     ; 0316
                00000000G  FF  D4 00064         CLRL    @PHAN+40                  ; 0317
   00000000G EF 00000000G  EF  D0 0006A         MOVL    TSF, FOHTSF               ; 0319
   00000000G EF 00000000G  EF  D0 00075         MOVL    MRA, FOHMRA               ; 0320
   00000000G EF 00000000G  EF  9E 00080         MOVAB   FOOTSF, TSF               ; 0321
   00000000G EF 00000000G  EF  9E 0008B         MOVAB   FOOMRA, MRA               ; 0322
                           56  D4 00096         CLRL    FOO_FIRST_XTN             ; 0323
                           53  D4 00098         CLRL    MAP_FIRST_XTN             ; 0324
             54            01  CE 0009A         MNEGL   #1, I                     ; 0359
                           27  11 0009D         BRB     7$                        ;
50           54        08  AB  C1 0009F 5$:     ADDL3   FNCT+12, I, R0            ; 0360
             58 00000000GEF40  D0 000A4         MOVL    FNISIZ[R0], R8            ;
                           55  D4 000AC         CLRL    J                         ;
                         0080  31 000AE         BRW     11$                       ;
                           08  DD 000B1 6$:     PUSHL   #8                        ; 0364
   00000000G EF            01  FB 000B3         CALLS   #1, FOOFIL                ;
             59            50  D0 000BA         MOVL    R0, STATUS                ;
                           52  D4 000BD         CLRL    R2                        ; 0366
             01            59  D1 000BF         CMPL    STATUS, #1                ;
                           05  13 000C2         BEQL    8$                        ;
                           52  D6 000C4         INCL    R2                        ;
                         00D1  31 000C6 7$:     BRW     17$                       ; 0369
             02 00000000G  EF  D1 000C9 8$:     CMPL    FOOREC, #2                ; 0372
                           61  12 000D0         BNEQ    12$                       ;
                           09  DD 000D2         PUSHL   #9                        ; 0377
   00000000G EF            01  FB 000D4         CALLS   #1, FOOFIL                ;
             59            50  D0 000DB         MOVL    R0, STATUS                ;
             01            59  D1 000DE         CMPL    STATUS, #1                ; 0380
                           E3  12 000E1         BNEQ    7$                        ;
             51 00000000G  EF  D0 000E3         MOVL    FOOREC+12, R1             ; 0386
                           2A  15 000EA         BLEQ    10$                       ;
             50 00000000GFF41  DE 000EC         MOVAL   @FOOREC+16[R1], R0        ; 0394
                       FC  A0  D5 000F4         TSTL    -4(R0)                    ;
                           1D  13 000F7         BEQL    10$                       ;
                           56  D5 000F9         TSTL    FOO_FIRST_XTN             ; 0398
                           0B  12 000FB         BNEQ    9$                        ;
             53 00000000G  EF  D0 000FD         MOVL    GCA+168, MAP_FIRST_XTN    ; 0403
             56        FC  A0  D0 00104         MOVL    -4(R0), FOO_FIRST_XTN     ; 0404
          FC A0 00000000G  EF  D0 00108 9$:     MOVL    GCA+168, -4(R0)           ; 0413
```

```
                                          00000000G  EF D6 00110          INCL    GCA+168                                         ; 0414
                                          00000000G  EF DD 00116 10$:     PUSHL   FOOREC+4                                        ; 0420
                                                     51 DD 0011C          PUSHL   R1                                              ; 0419
                                          00000000G  EF DD 0011E          PUSHL   FOOREC+16                                       ;
                                          00000000G  EF DD 00124          PUSHL   FOOREC+8                                        ;
                            00000000G  EF            04 FB 0012A          CALLS   #4, OUTPAS                                      ;
                                                     61 11 00131 11$:     BRB     16$                                             ;
                                                     03 DD 00133 12$:     PUSHL   #3                                              ; 0425
                            00000000G  EF            01 FB 00135          CALLS   #1, FOOFIL                                      ;
                                       5B            52 E8 0013C          BLBS    R2, 17$                                         ; 0428
                                       50 00000000G  EF D0 0013F          MOVL    TSF, R0                                         ; 0463
                                       51       38   A0 D0 00146          MOVL    56(R0), R1                                      ;
                                                     3F 13 0014A          BEQL    15$                                             ;
                                       26       14   A0 E9 0014C          BLBC    20(R0), 14$                                     ; 0467
                                                     56 D5 00150          TSTL    FOO_FIRST_XTN                                   ; 0472
                                                     0A 12 00152          BNEQ    13$                                             ;
                                       53 00000000G  EF D0 00154          MOVL    GCA+168, MAP_FIRST_XTN                          ; 0477
                                       56            51 D0 0015B          MOVL    R1, FOO_FIRST_XTN                               ; 0478
                                  38   A0 00000000G  EF D0 0015E 13$:     MOVL    GCA+168, 56(R0)                                 ; 0487
                                  3C   A0 00000000G  EF D0 00166          MOVL    GCA+168, 60(R0)                                 ; 0488
                                          00000000G  EF D6 0016E          INCL    GCA+168                                         ; 0489
                                                     15 11 00174          BRB     15$                                             ; 0467
                         52            51            56 C3 00176 14$:     SUBL3   FOO_FIRST_XTN, R1, XTN_OFFSET                   ; 0508
                                       52            53 C0 0017A          ADDL2   MAP_FIRST_XTN, R2                               ; 0514
                         51       3C   A0            51 C3 0017D          SUBL3   R1, 60(R0), R1                                  ;
                    3C   A0            51            52 C1 00182          ADDL3   R2, R1, 60(R0)                                  ;
                                  38   A0            52 D0 00187          MOVL    R2, 56(R0)                                      ; 0517
                                                     7E D4 0018B 15$:     CLRL    -(SP)                                           ; 0527
                            00000000G  EF            01 FB 0018D          CALLS   #1, OUTLIN                                      ;
         FF17            55            01            58 F1 00194 16$:     ACBL    R8, #1, J, 6$                                   ; 0360
                         02            54            57 F2 0019A 17$:     AOBLSS  COUNT, I, 18$                                   ;
                                                     03 11 0019E          BRB     19$                                             ;
                                                   FEFC 31 001A0 18$:     BRW     5$                                              ;
                            00000000G  EF 00000000G  EF D0 001A3 19$:     MOVL    FOHTSF, TSF                                     ; 0532
                            00000000G  EF 00000000G  EF D0 001AE          MOVL    FOHMRA, MRA                                     ; 0533
                            00000000G  FF            5A D0 001B9          MOVL    HOLD_PAGING, @PHAN+40                           ; 0534
                                       6B            57 C2 001C0          SUBL2   COUNT, FNCT+4                                   ; 0535
                                  FC   AB            57 C2 001C3          SUBL2   COUNT, FNCT                                     ; 0536
                                  08   AB            57 C0 001C7          ADDL2   COUNT, FNCT+12                                  ; 0537
                                                14   AB D4 001CB          CLRL    FNCT+24                                         ; 0538
                                                FC   AB D5 001CE          TSTL    FNCT                                            ; 0541
                                                     17 14 001D1          BGTR    20$                                             ;
                                                     05 DD 001D3          PUSHL   #5                                              ; 0545
                            00000000G  EF            01 FB 001D5          CALLS   #1, FOOFIL                                      ;
                                                     06 DD 001DC          PUSHL   #6                                              ; 0546
                            00000000G  EF            01 FB 001DE          CALLS   #1, FOOFIL                                      ;
                                                     6B D4 001E5          CLRL    FNCT+4                                          ; 0548
                                                08   AB 7C 001E7          CLRQ    FNCT+12                                         ; 0547
                                                        04 001EA 20$:     RET                                                     ; 0552

; Routine Size:  491 bytes,    Routine Base:  $CODE$ + 0000


;  426            0553  1
;  427            0554  1 END                                             !End of module
;  428            0555  0 ELUDOM
;                                    PSECT SUMMARY
;
;       Name                     Bytes                        Attributes
;
;   $PLIT$                                  8  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                                491  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                             Library Statistics
;
;                                              -------- Symbols --------        Pages        Processing
;       File                                   Total    Loaded   Percent       Mapped       Time
;
;   _$255$DUA28:[SYSLIB]XPORT.L32;1              590         0         0          252         00:00.2
;   _$255$DUA28:[RUNOFF.SRC]DSRLIB.L32;1        1248        37         2           86         00:00.3




;                                     COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:FOOOUT/OBJ=OBJ$:FOOOUT MSRC$:FOOOUT/UPDATE=(ENH$:FOOOUT)

; Size:           491 code + 8 data bytes
; Run Time:          00:10.3
; Elapsed Time:      00:31.4
; Lines/CPU Min:      3229
; Lexemes/CPU-Min: 15660
; Memory Used:  119 pages
; Compilation Complete
```

0341 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY